AG44 : Ariadne's thread

Guillaume OBERLE

25 October 2013

1 Name of the problem

A level is defined by the possibility to go from one place to any place in this level. So, we deduce that in this game, each level represent a strongly connected component.

2 Data Structures

To solve this problem, I chose the adjacency list to represent the graph. The implementation is in Python, and dictionaries are perfect and nice to use in order to represent graphs. Here is an example of an adjacency list :

{1: [2], 2: [3, 4], 3: [2, 5, 7], 4: [5, 8], 5: [4, 8], 6: [2, 8], 7: [6, 8], 8: []}

3 Algorithms

To determine the strongly connected components, I chose to use the Tarjan Algorithm. I could have use the + and - algorithm that we have done in course, but as I have understood how this algorithm works, I preferred to discover and try to implant something else.

To determine the longest path, I have used the Topological sorting algorithm. Once I have had the topological order, I have process one by one all vertices in topological order. For every vertex being processed, I updated distances of its adjacent using distance of current vertex.

4 Example

Strongly connected components for the matrix given in the subject is :

[[13, 12, 11, 10], [6, 5, 7], [9, 8], [3, 4, 2, 1]]

The value for the reduced matrix is :

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \end{bmatrix}$$

And the longest path is :

[4, 2, 1]